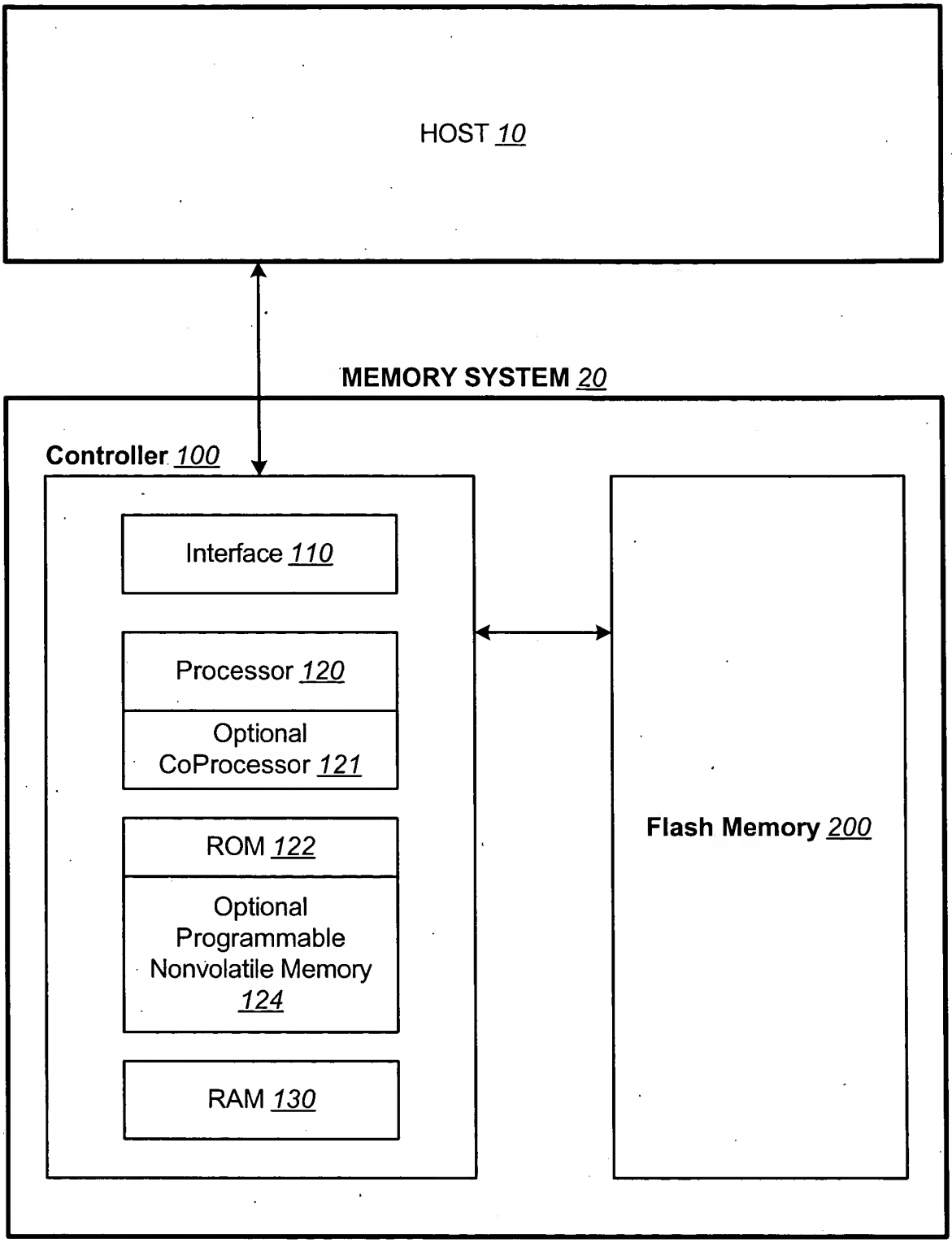

HOST 10
MEMORY SYSTEM 20
Controller 100
Interface 110
Processor 120
Optional CoProcessor 121
ROM 122
Optional Programmable Nonvolatile Memory 124
RAM 130
Flash Memory 200

FIG. 1

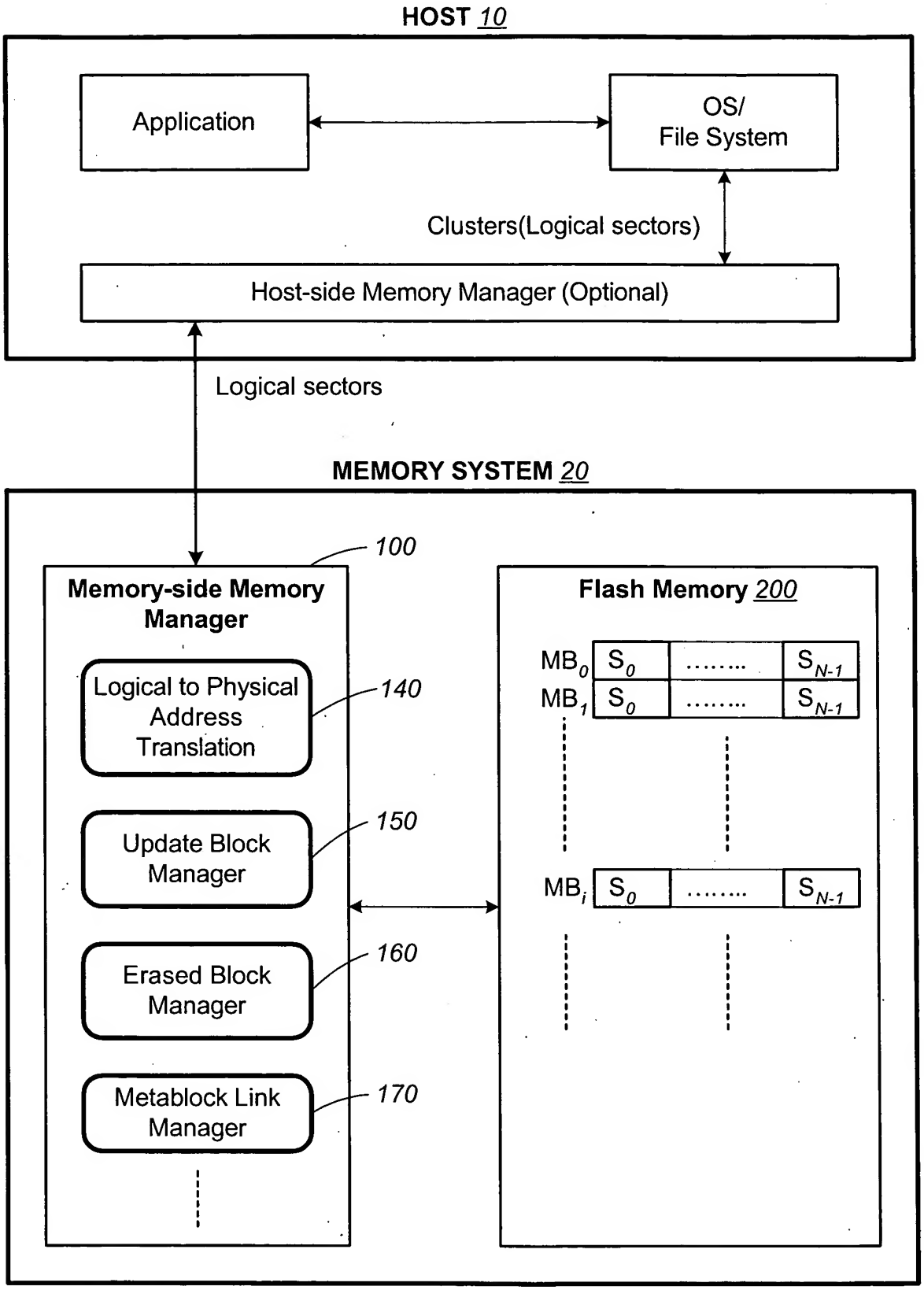

HOST 10
Application
OS/
File System
Clusters(Logical sectors)
Host-side Memory Manager (Optional)
Logical sectors
MEMORY SYSTEM 20
100
Memory-side Memory Manager
Logical to Physical Address Translation
140
Update Block Manager
150
Erased Block Manager
160
Metablock Link Manager
170
Flash Memory 200
MB0
S0
SN-1
MB1
MBi

FIG. 2

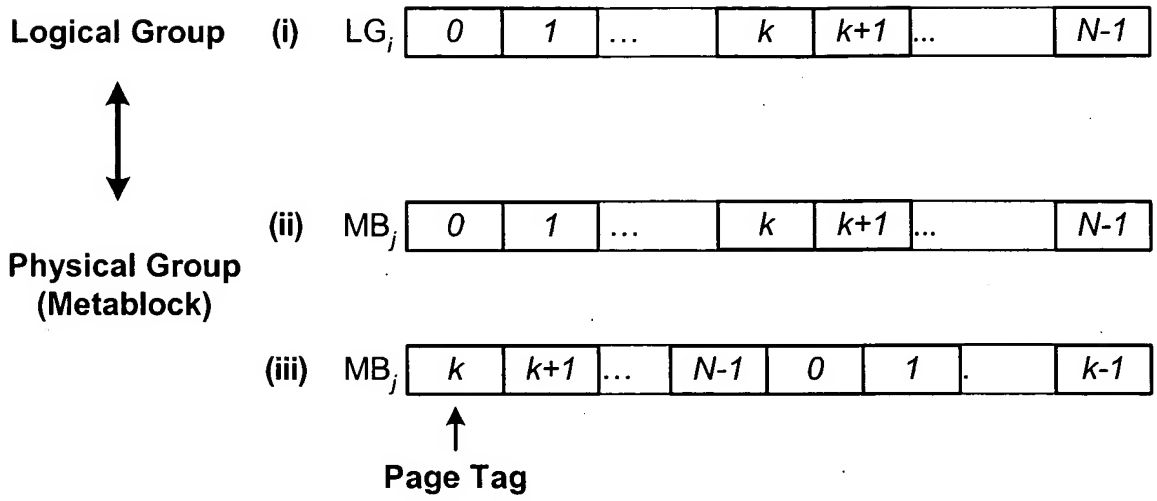

Logical Group
(i)
LG$_i$
0
1
...
k
k+1
...
N-1
Physical Group
(Metablock)
(ii)
MB$_j$
0
1
...
k
k+1
...
N-1
(iii)
MB$_j$
k
k+1
...
N-1
0
1
.
k-1
Page Tag

FIG. 3A

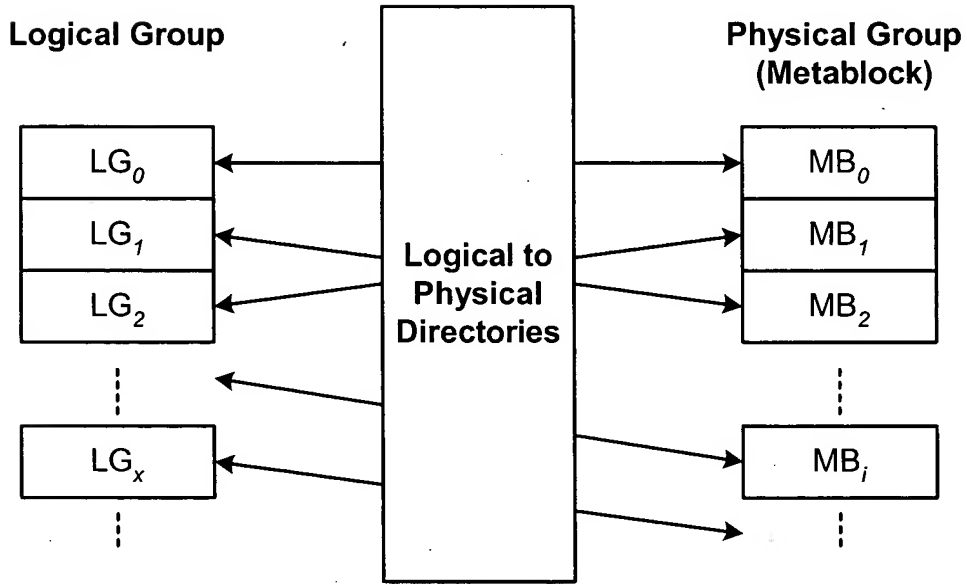
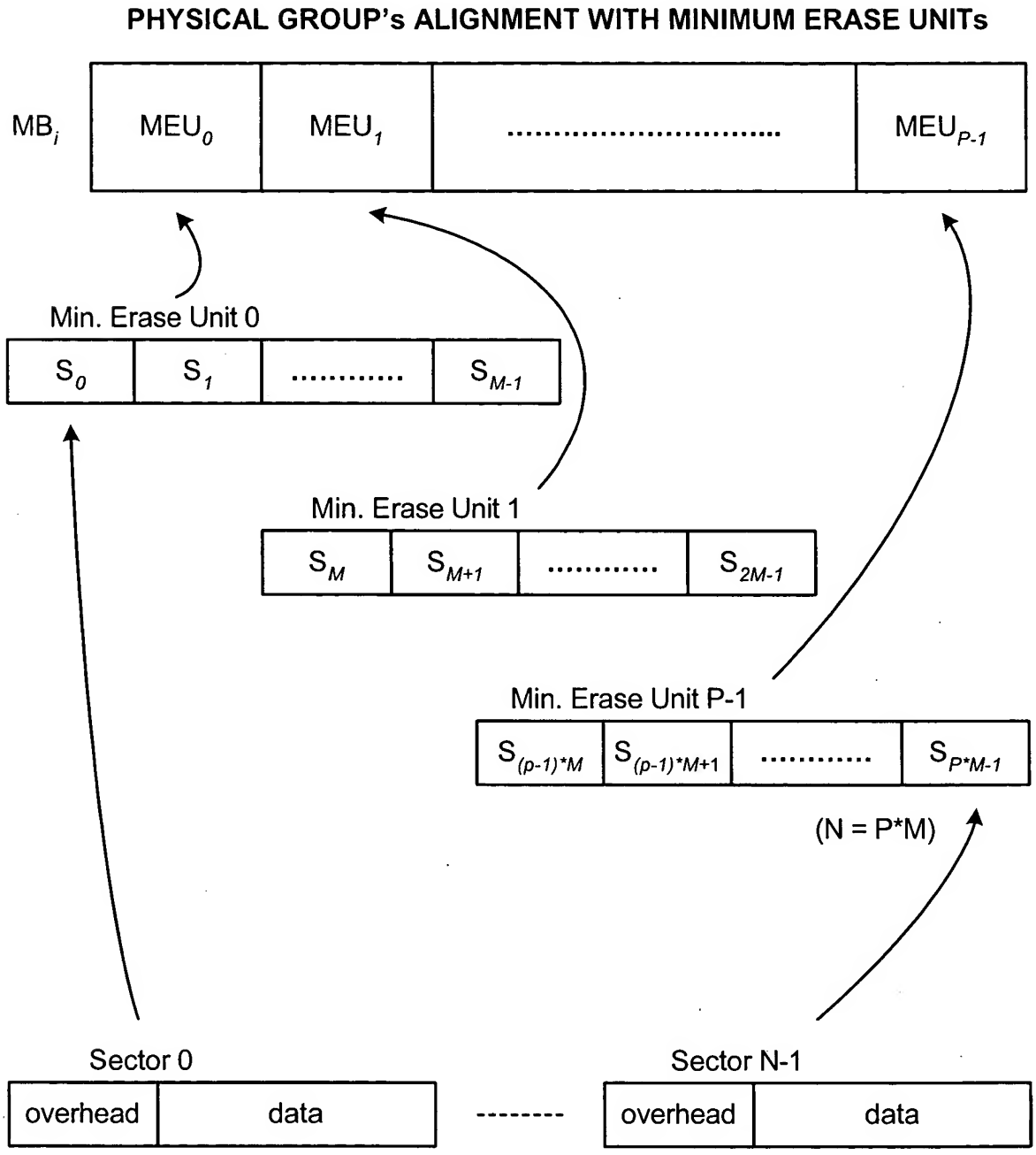

Logical Group
Physical Group
(Metablock)
LG$_0$
LG$_1$
LG$_2$
LG$_x$
Logical to
Physical
Directories
MB$_0$
MB$_1$
MB$_2$
MB$_i$

FIG. 3B


PHYSICAL GROUP's ALIGNMENT WITH MINIMUM ERASE UNITs
MB$_i$
MEU$_0$
MEU$_1$
.....
MEU$_{P-1}$
Min. Erase Unit 0
S$_0$
S$_1$
.....
S$_{M-1}$
Min. Erase Unit 1
S$_M$
S$_{M+1}$
.....
S$_{2M-1}$
Min. Erase Unit P-1
S$_{(p-1)*M}$
S$_{(p-1)*M+1}$
.....
S$_{P*M-1}$
(N = P*M)
Sector 0
overhead
data

Sector N-1
overhead
data

FIG. 4

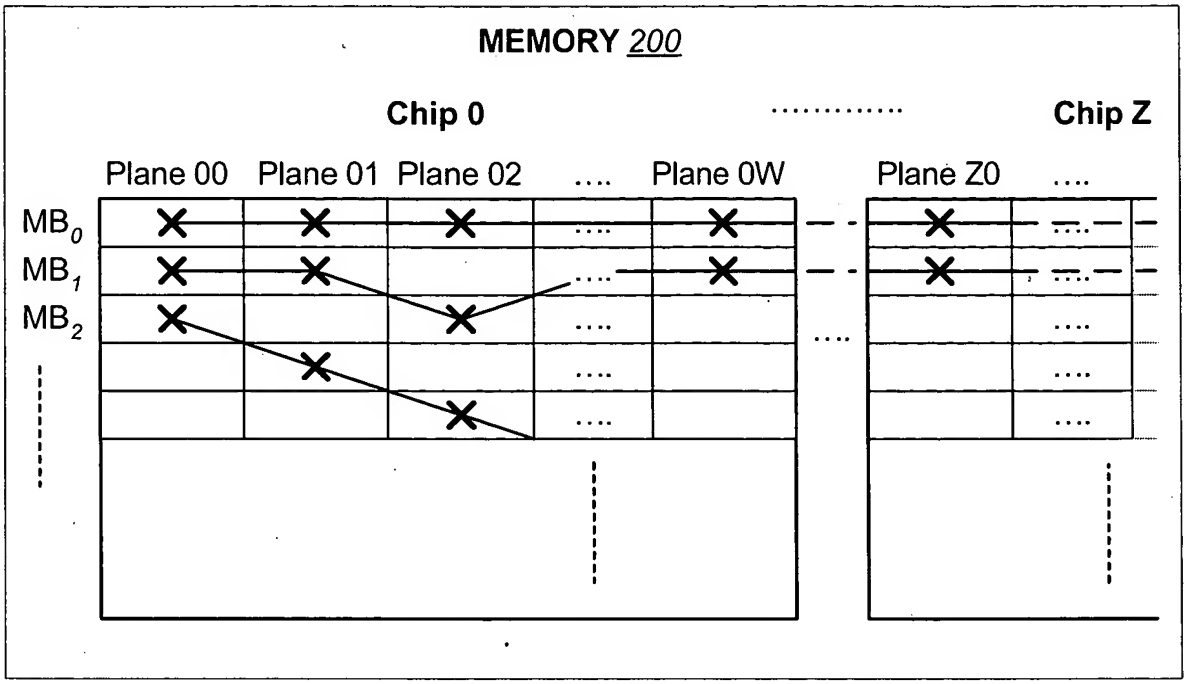


FIG. 5A

FIG. 5B

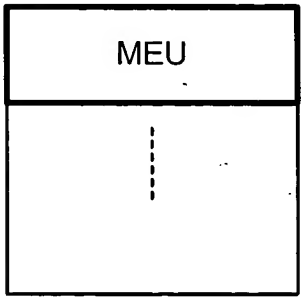


FIG. 5C

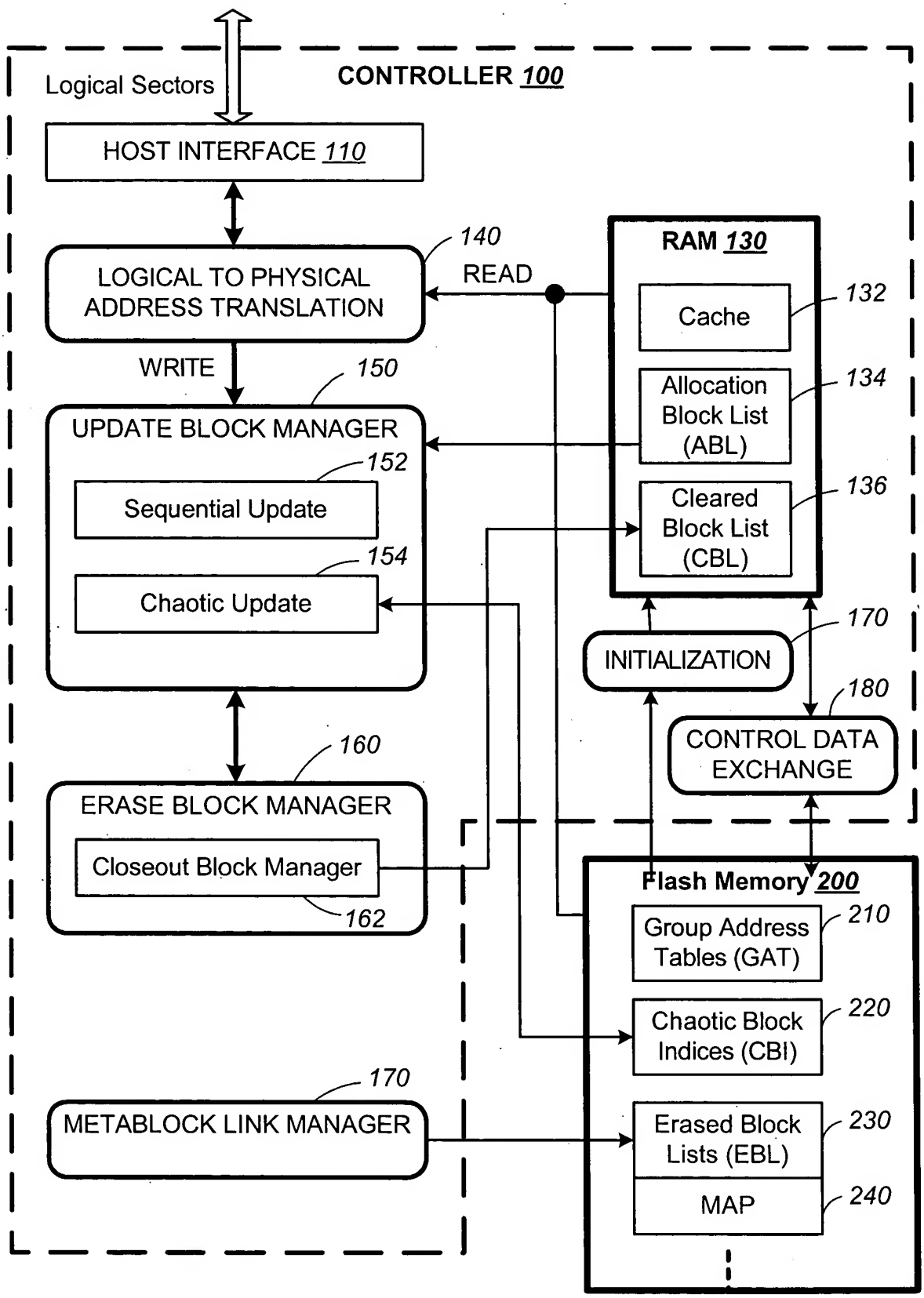
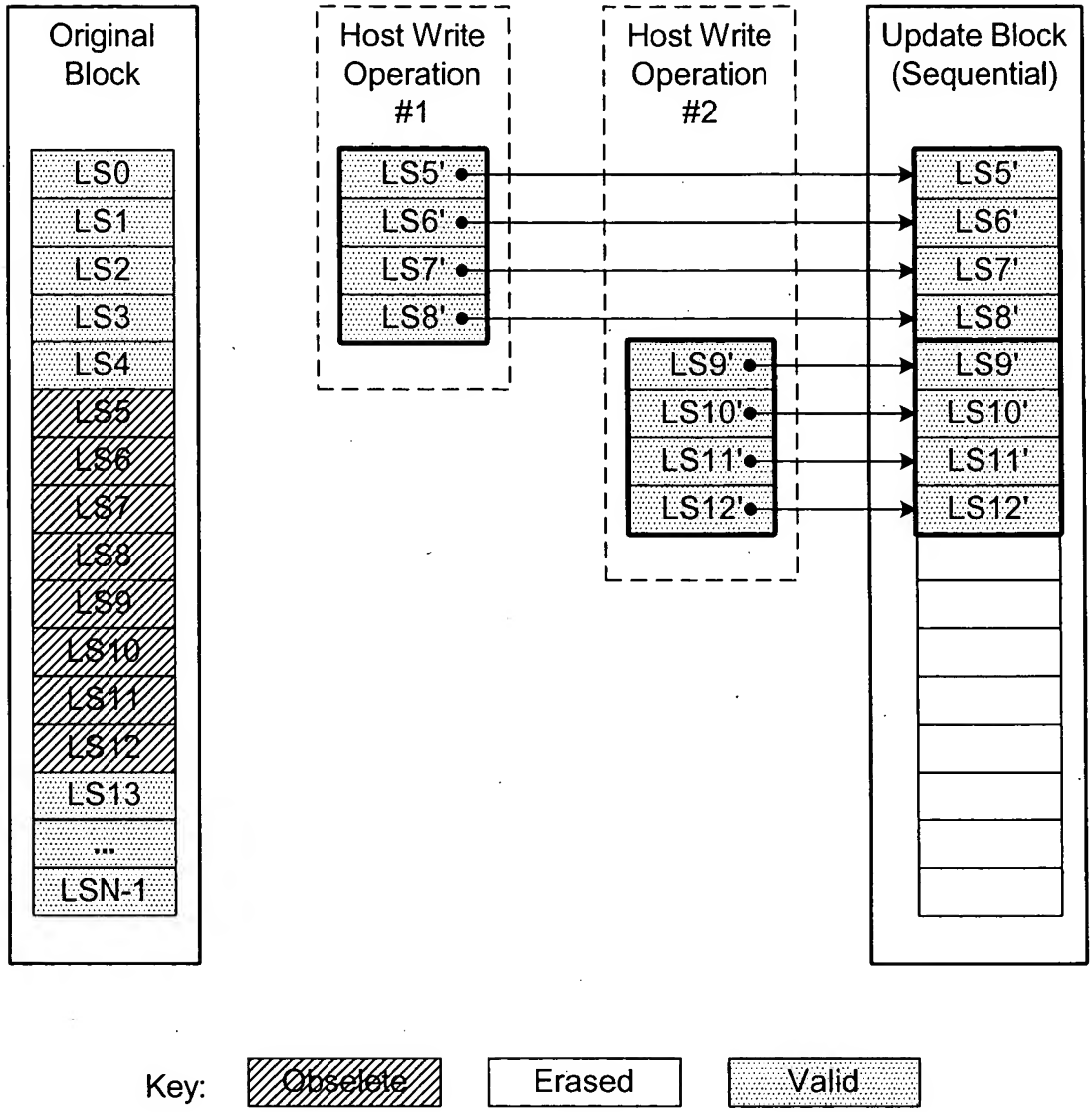
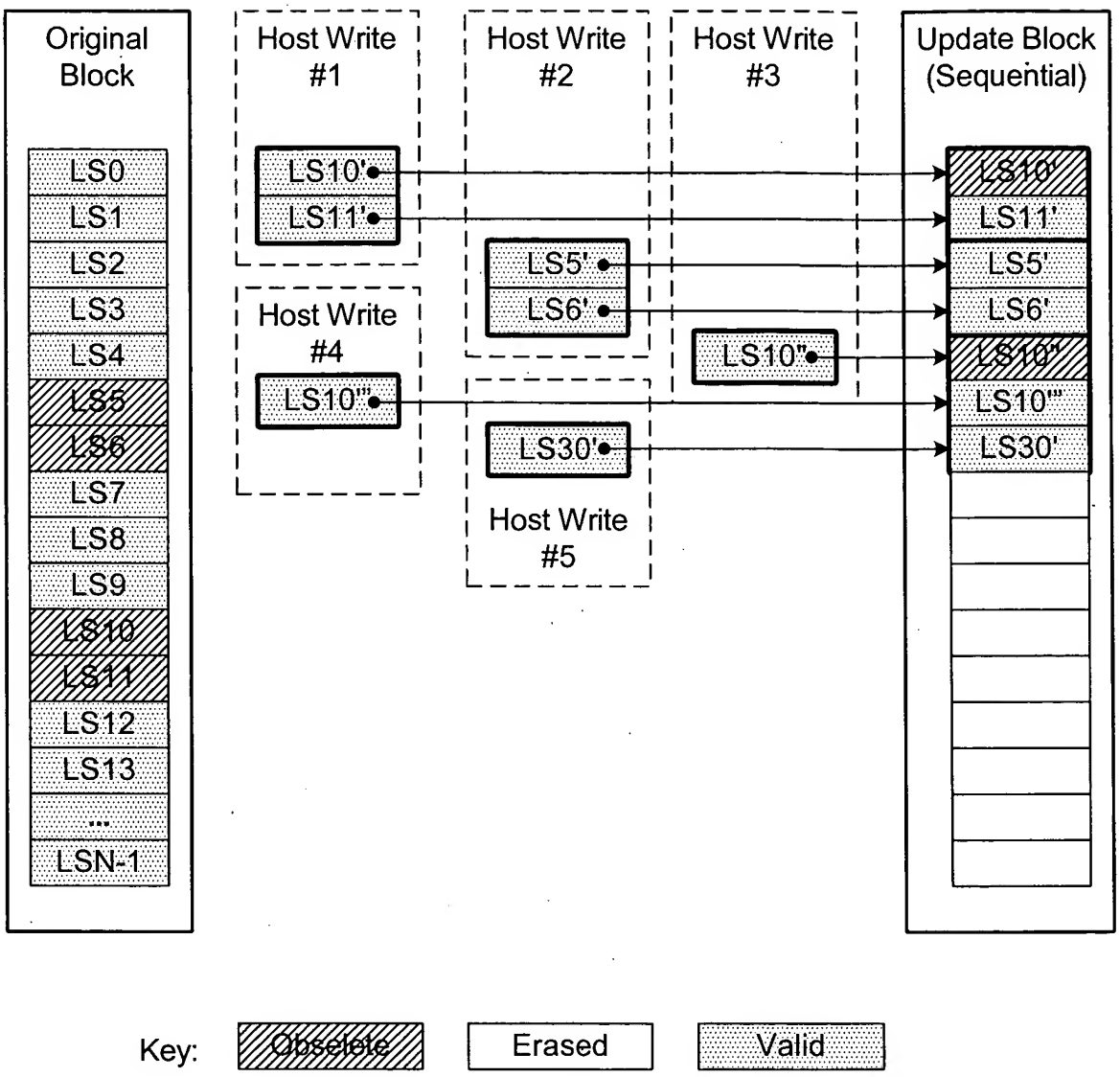

Logical Sectors
CONTROLLER 100
HOST INTERFACE 110
LOGICAL TO PHYSICAL ADDRESS TRANSLATION
140
READ
WRITE
150
UPDATE BLOCK MANAGER
152
Sequential Update
154
Chaotic Update
160
ERASE BLOCK MANAGER
Closeout Block Manager
162
170
METABLOCK LINK MANAGER
RAM 130
Cache
132
Allocation Block List (ABL)
134
Cleared Block List (CBL)
136
170
INITIALIZATION
180
CONTROL DATA EXCHANGE
Flash Memory 200
Group Address Tables (GAT)
210
Chaotic Block Indices (CBI)
220
Erased Block Lists (EBL)
230
MAP
240

FIG. 6



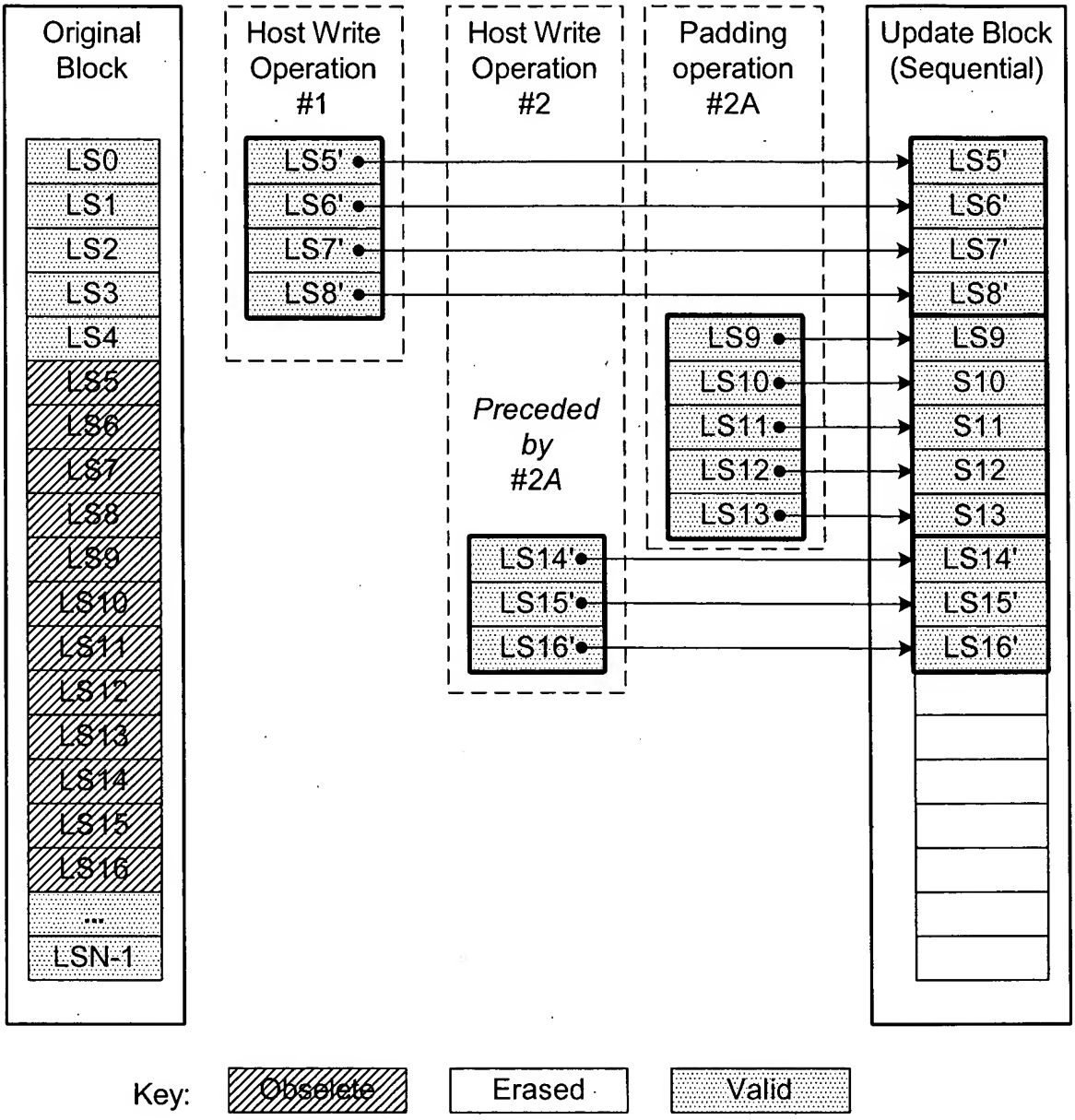
SEQUENTIAL UPDATE
Example

FIG. 7A



CHAOTIC UPDATE (NONSEQUENTIAL) Example

FIG. 7B



FORCED SEQUENTIAL UPDATE Example

FIG. 8

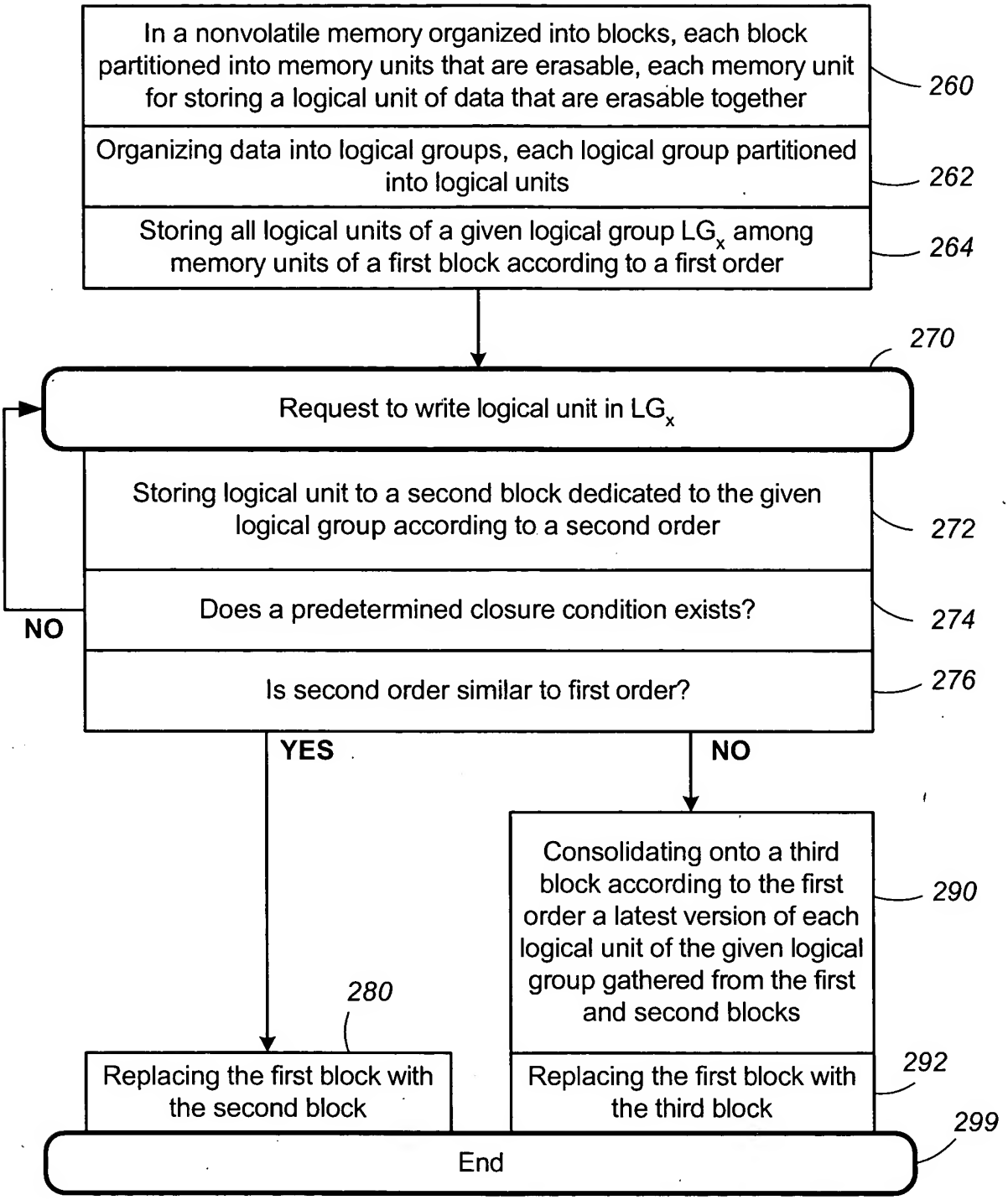

In a nonvolatile memory organized into blocks, each block partitioned into memory units that are erasable, each memory unit for storing a logical unit of data that are erasable together
260
Organizing data into logical groups, each logical group partitioned into logical units
262
Storing all logical units of a given logical group LG_x among memory units of a first block according to a first order
264
270
Request to write logical unit in LG_x
Storing logical unit to a second block dedicated to the given logical group according to a second order
272
Does a predetermined closure condition exists?
274
NO
Is second order similar to first order?
276
YES
NO
Consolidating onto a third block according to the first order a latest version of each logical unit of the given logical group gathered from the first and second blocks
290
280
Replacing the first block with the second block
Replacing the first block with the third block
292
End
299

FIG. 9

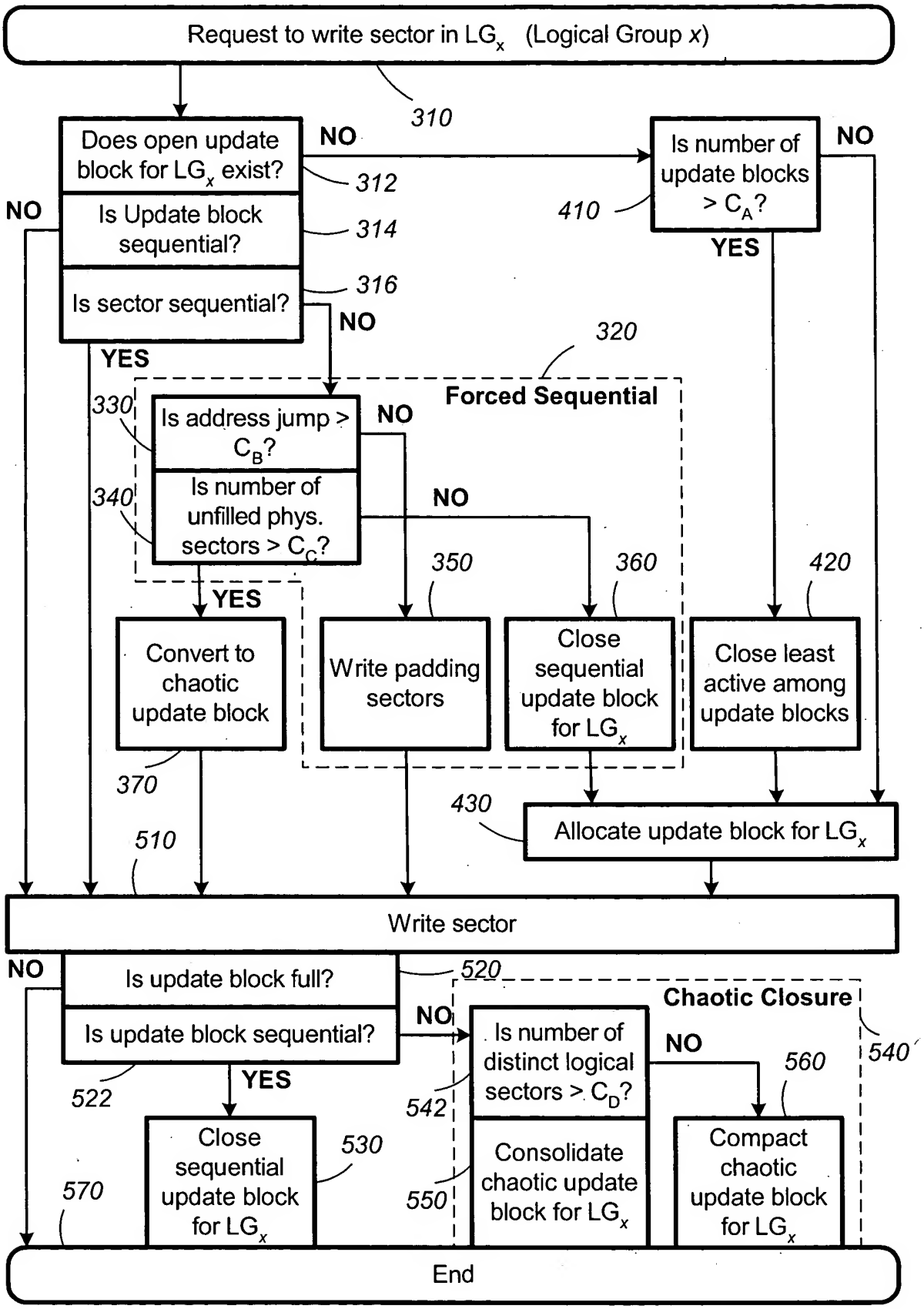

Request to write sector in LG$_x$ (Logical Group x)
310
Does open update block for LG$_x$ exist?
NO
312
Is Update block sequential?
314
Is sector sequential?
316
NO
YES
NO
Is number of update blocks > C$_A$?
410
NO
YES
320
Forced Sequential
330
Is address jump > C$_B$?
NO
340
Is number of unfilled phys. sectors > C$_C$?
NO
YES
350
360
420
Convert to chaotic update block
Write padding sectors
Close sequential update block for LG$_x$
Close least active among update blocks
370
430
Allocate update block for LG$_x$
510
Write sector
NO
Is update block full?
520
Is update block sequential?
NO
Chaotic Closure
540
YES
522
Is number of distinct logical sectors > C$_D$?
542
NO
560
Close sequential update block for LG$_x$
530
570
Consolidate chaotic update block for LG$_x$
550
Compact chaotic update block for LG$_x$
End

FIG. 10

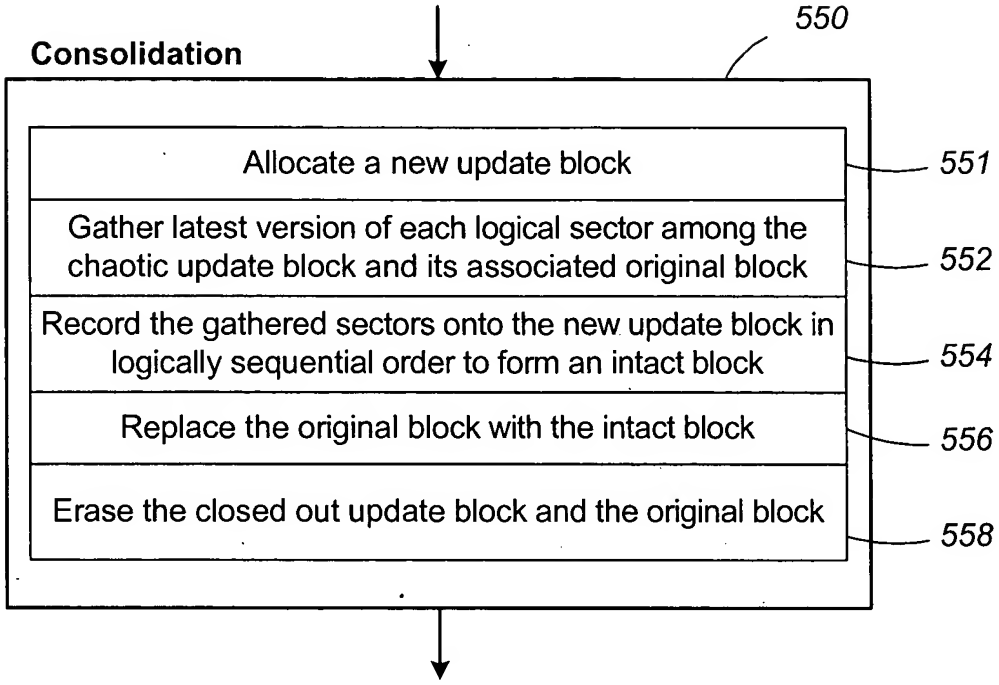


FIG. 11A

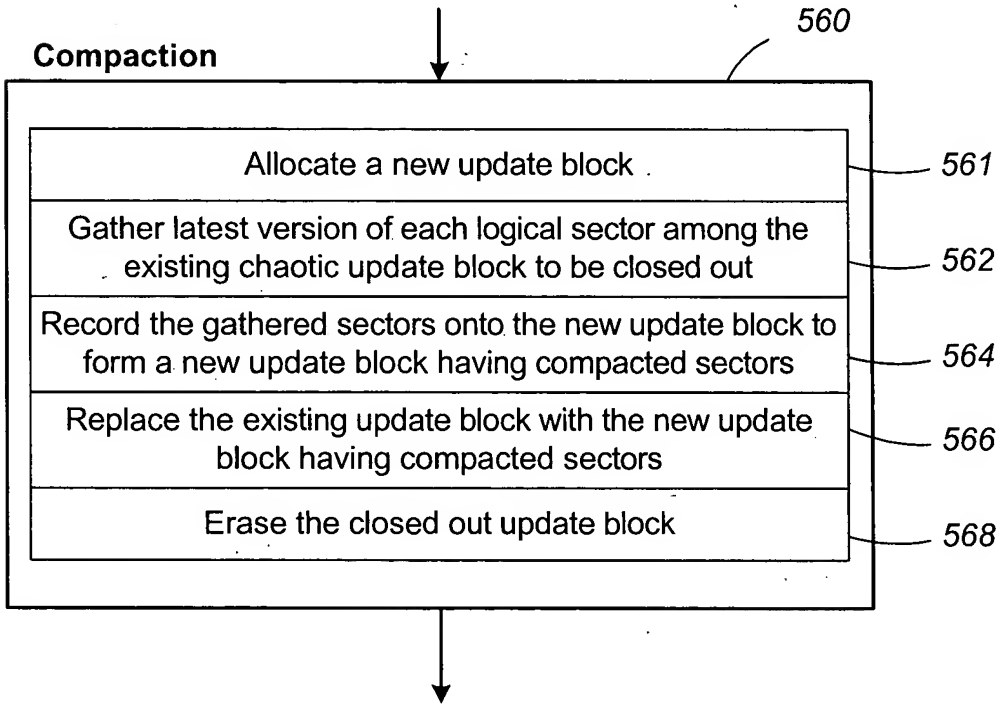
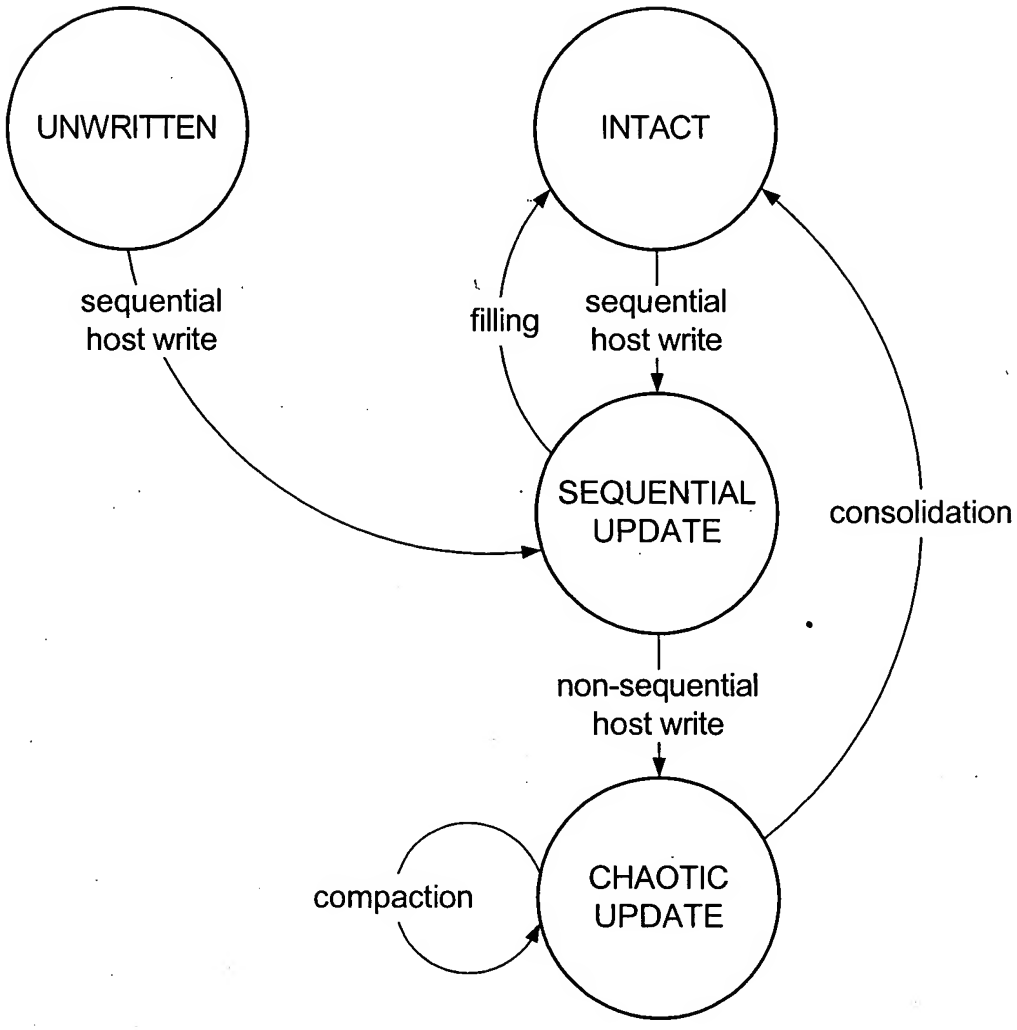


FIG. 11B

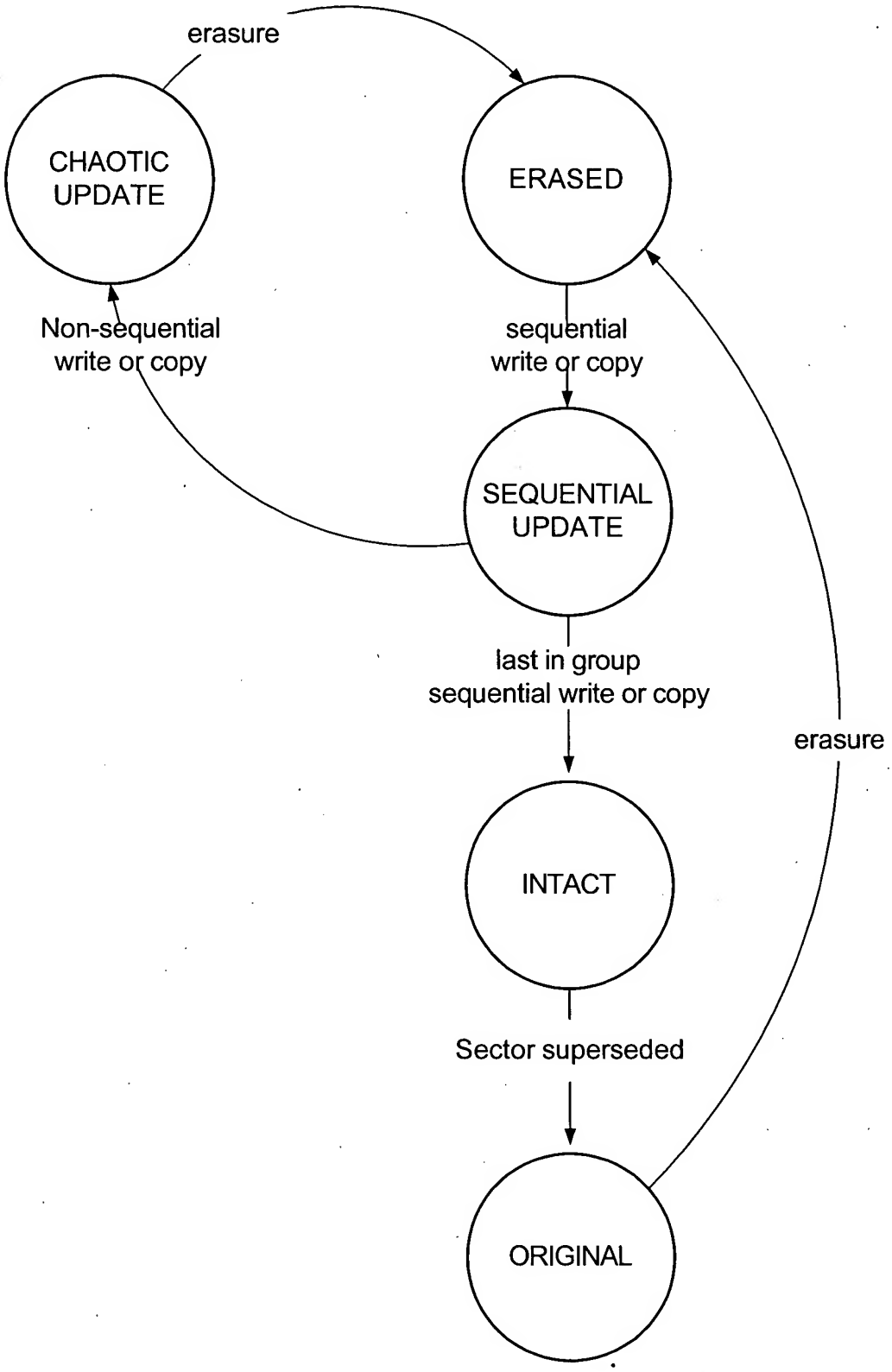


LOGICAL GROUP STATES

FIG. 12A

LOGICAL GROUP STATE	DESCRIPTION
Intact	Complete Logical Group in logically sequential order in a single metablock
Unwritten	No portion of the Logical Group has ever been written
Sequential Update	An update segment of the Logical Group is written to an update block without changing its existing sequential nature
Chaotic Update	An update segment of the Logical Group is written to an update block in logically non-sequential order

FIG. 12B

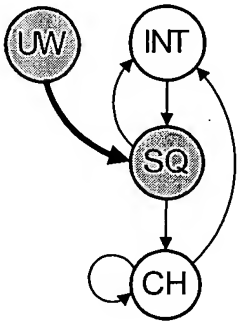
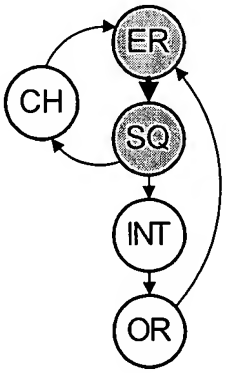
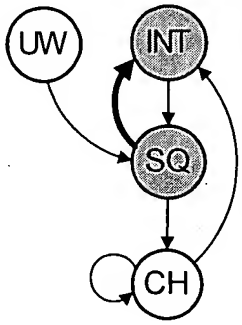
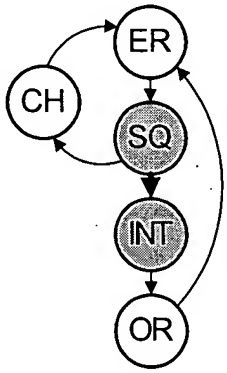


PHYSICAL METABLOCK STATES

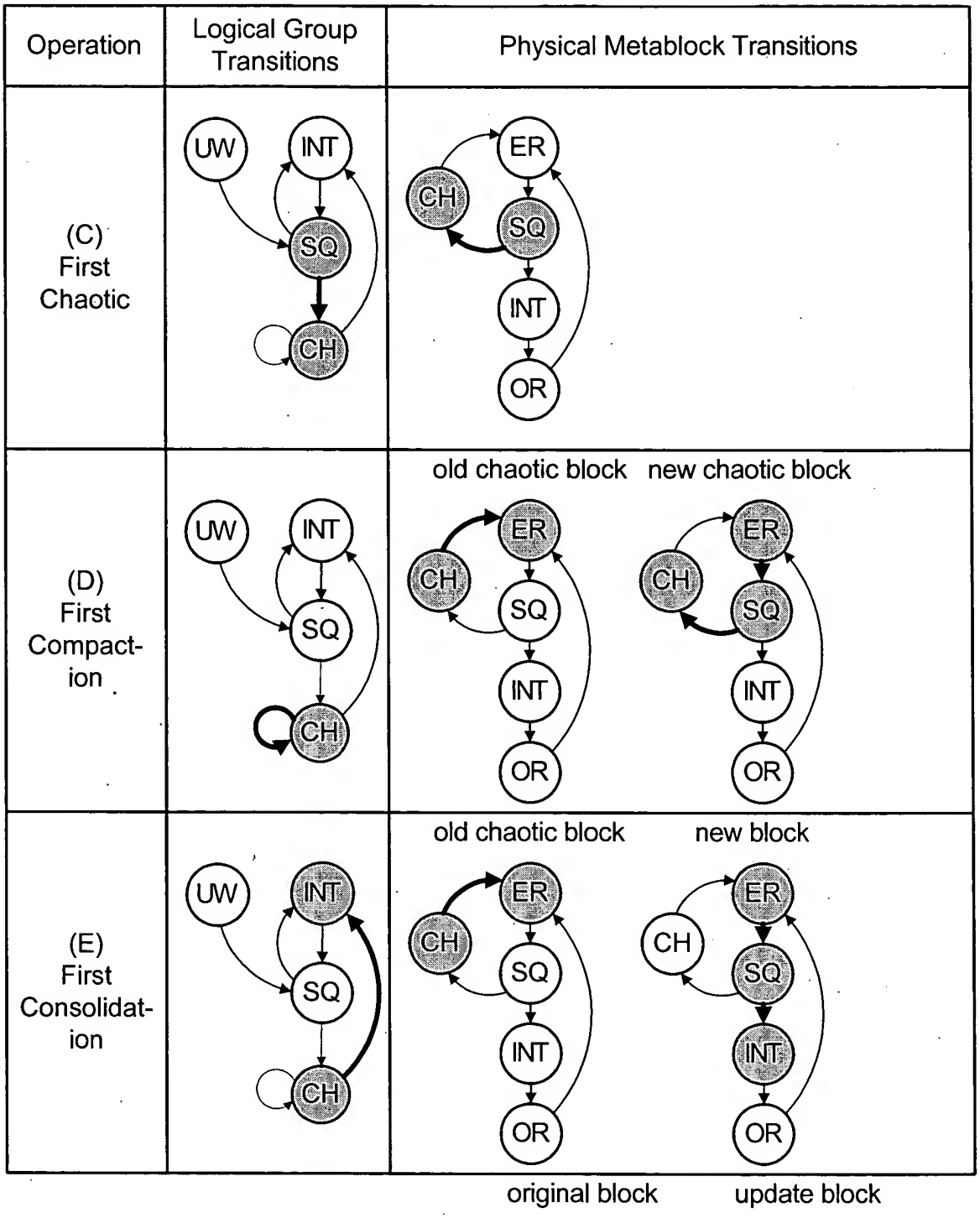
FIG. 13A

METABLOCK STATE	DESCRIPTION
Erased	All the sectors in the metablock are erased
Sequential Update	The metablock is partially written with sectors in logically sequential order, possibly using page tag. All the sectors belong to the same logical group.
Non-sequential (Chaotic) Update	The metablock is partially or fully written with sectors in logically non-sequential order. Any sector can be written more than once. All sectors belong to the same logical group
Intact	The metablock is fully written in logically sequential order, possibly using page tag
Original	The metablock was previously Intact but at least one sector has been made obsolete by a host data update

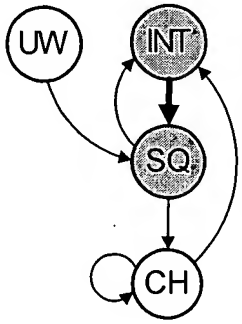
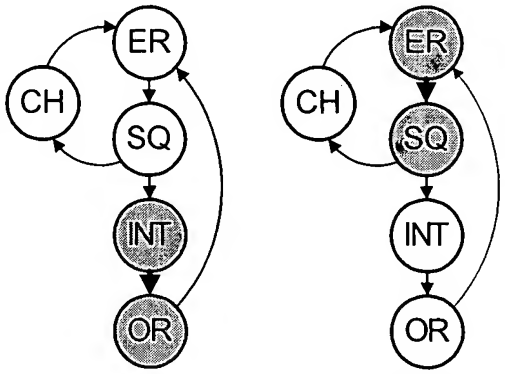
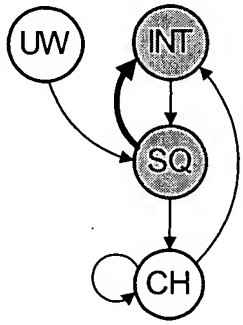
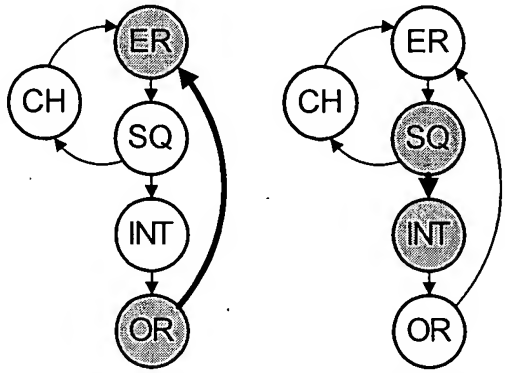
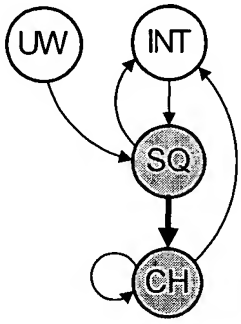
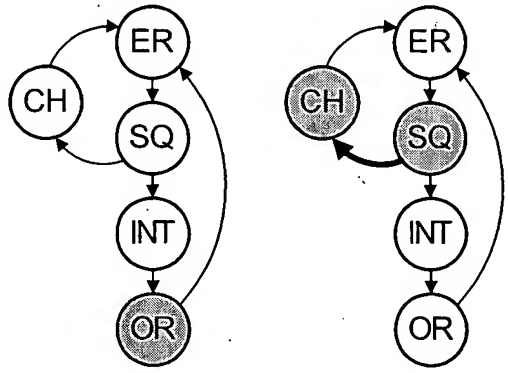
FIG. 13B

Operation	Logical Group Transitions	Physical Metablock Transitions
(A) First Write		
(B) First Intact		

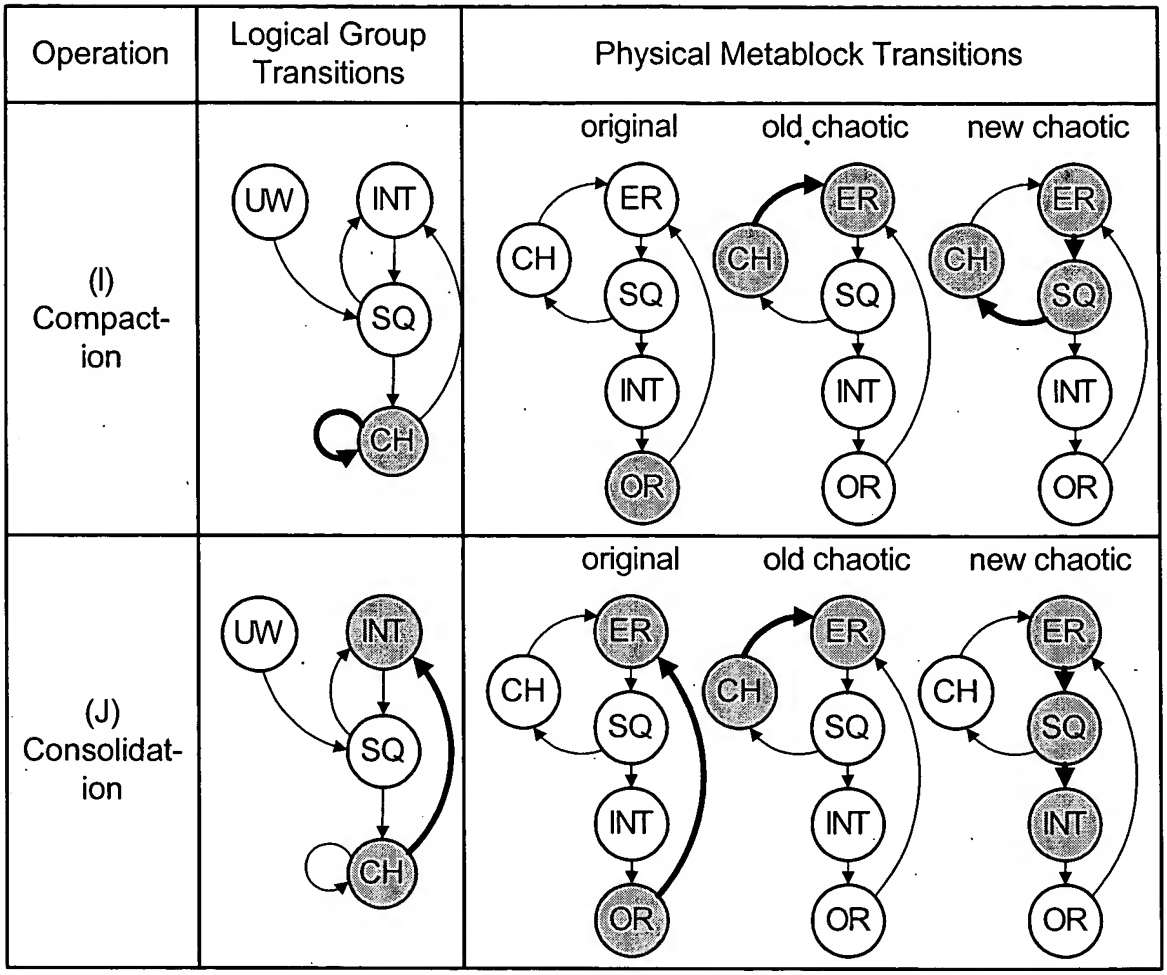
FIGs. 14(A)-14(B)


Operation
Logical Group Transitions
Physical Metablock Transitions
(C) First Chaotic
UW
INT
SQ
CH
ER
CH
SQ
INT
OR
(D) First Compact-ion
old chaotic block
new chaotic block
(E) First Consolidat-ion
old chaotic block
new block
original block
update block

FIGs. 14(C)-14(E)

Operation	Logical Group Transitions	Physical Metablock Transitions
(F) Sequential Write		
(G) Sequential Fill		
(H) Non-sequential Write		

FIGs. 14(F)-14(H)


Operation
Logical Group Transitions
Physical Metablock Transitions
original
old chaotic
new chaotic
(I) Compact-ion
(J) Consolidat-ion
UW
INT
SQ
CH
ER
OR

FIGs. 14(I)-14(J)

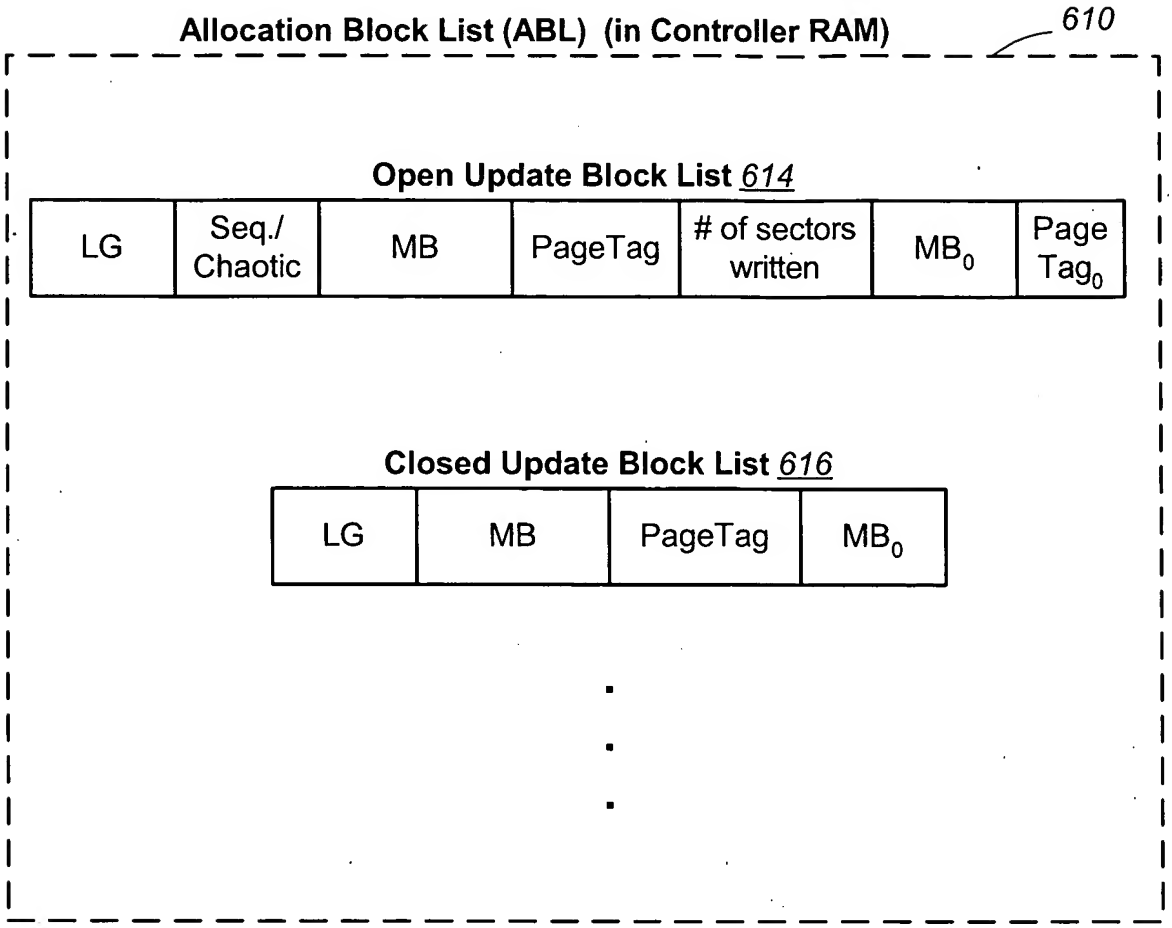

Allocation Block List (ABL) (in Controller RAM)
610
Open Update Block List 614
LG
Seq./ Chaotic
MB
PageTag
# of sectors written
MB0
Page Tag0
Closed Update Block List 616
LG
MB
PageTag
MB0

FIG. 15

Chaotic Block Index (CBI) Sector

Chaotic Block Index	Chaotic Block Info	CBI Sector Index

FIG. 16A

CBI Block *620*

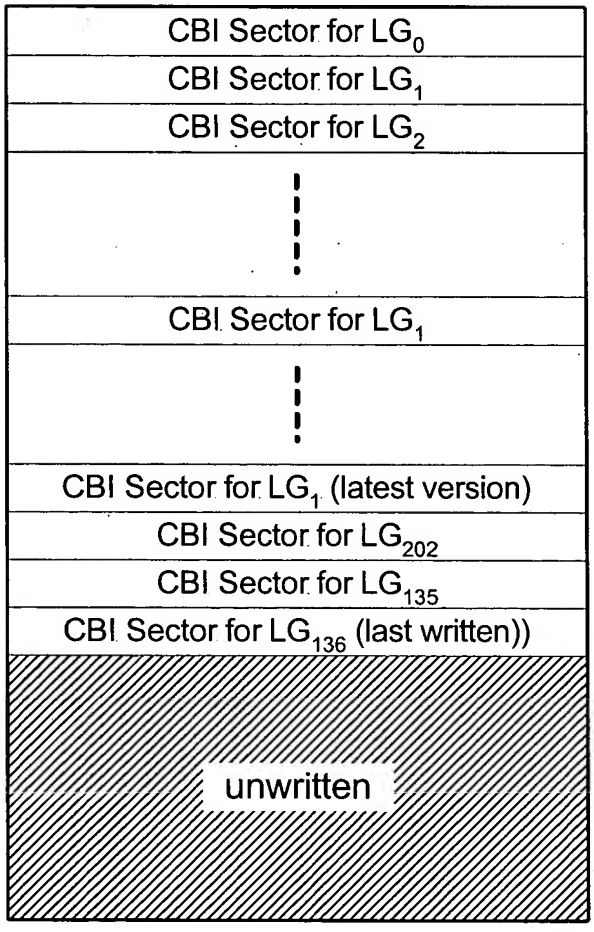


FIG. 16B

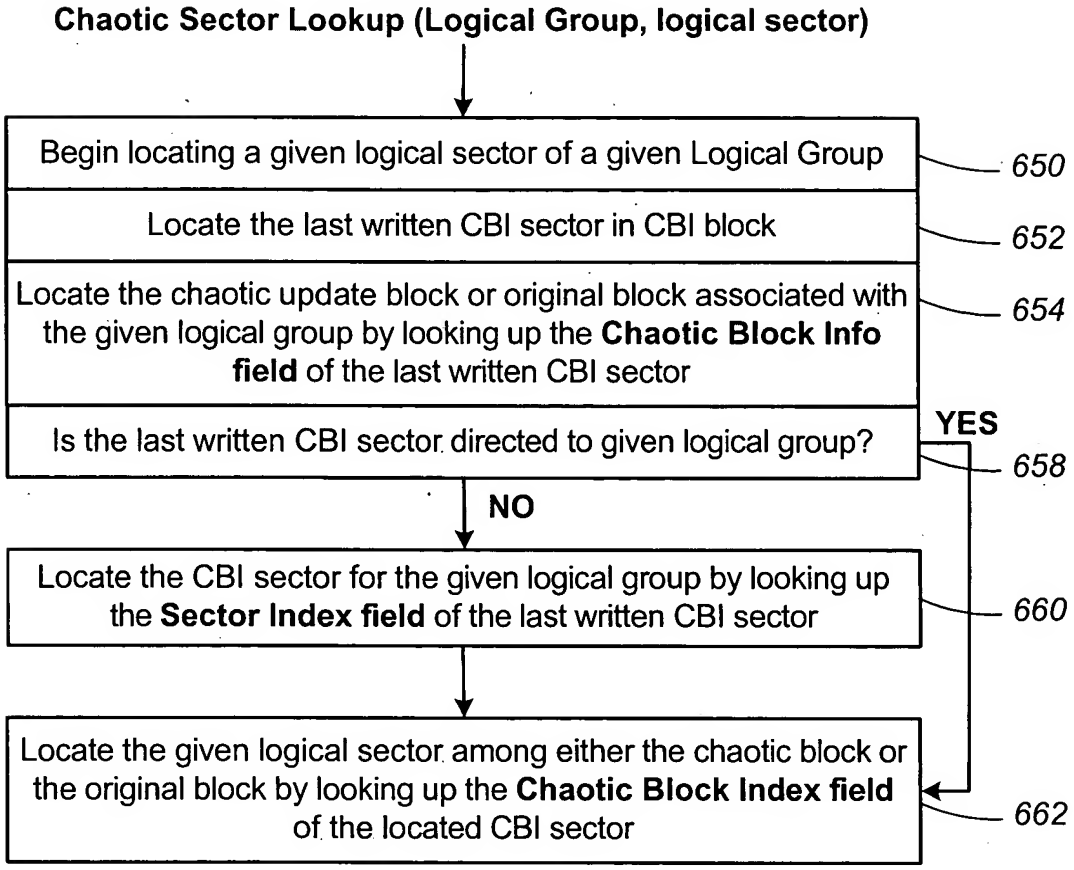

Chaotic Sector Lookup (Logical Group, logical sector)
Begin locating a given logical sector of a given Logical Group
650
Locate the last written CBI sector in CBI block
652
Locate the chaotic update block or original block associated with the given logical group by looking up the **Chaotic Block Info field** of the last written CBI sector
654
Is the last written CBI sector directed to given logical group?
YES
658
NO
Locate the CBI sector for the given logical group by looking up the **Sector Index field** of the last written CBI sector
660
Locate the given logical sector among either the chaotic block or the original block by looking up the **Chaotic Block Index field** of the located CBI sector
662

FIG. 16C

Chaotic Sector Lookup (Logical Group, Subgroup, logical sector)

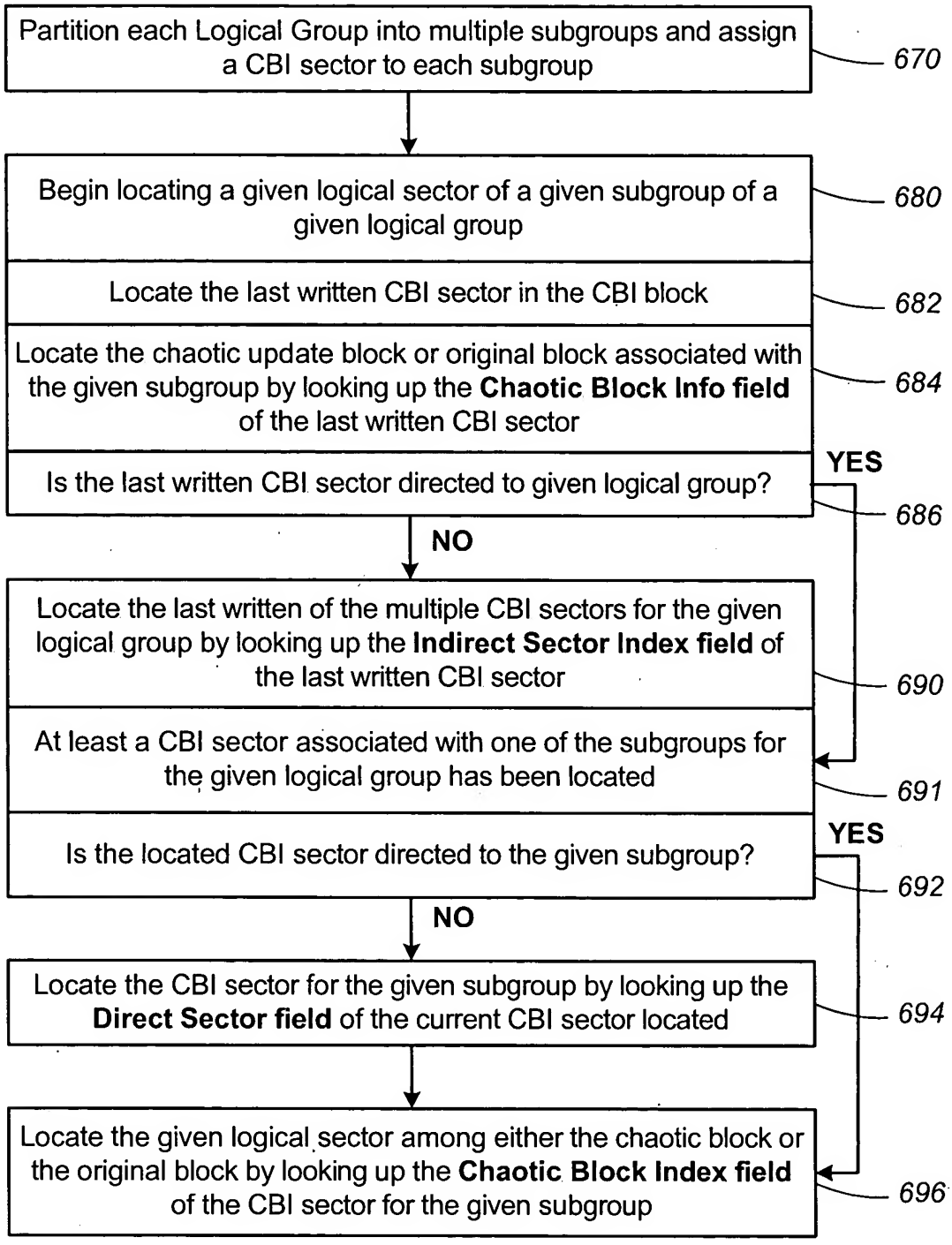
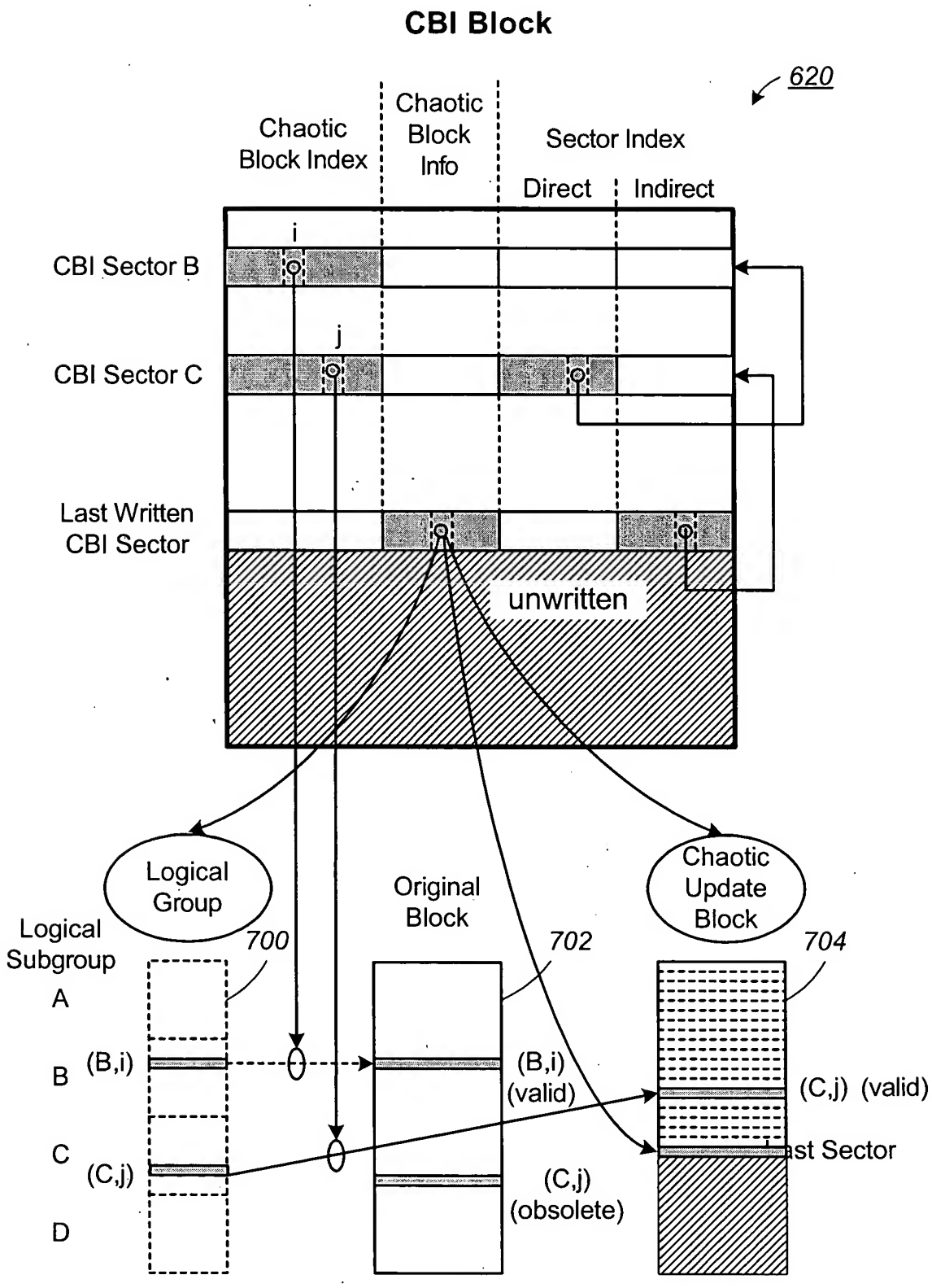


FIG. 16D



Chaotic Block Indexing Structure

FIG. 16E

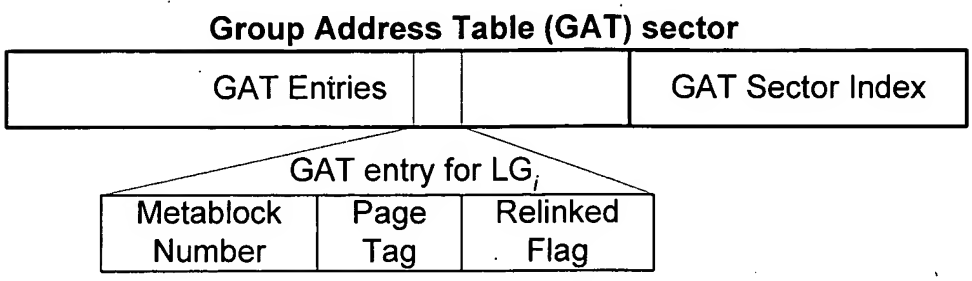


FIG. 17A

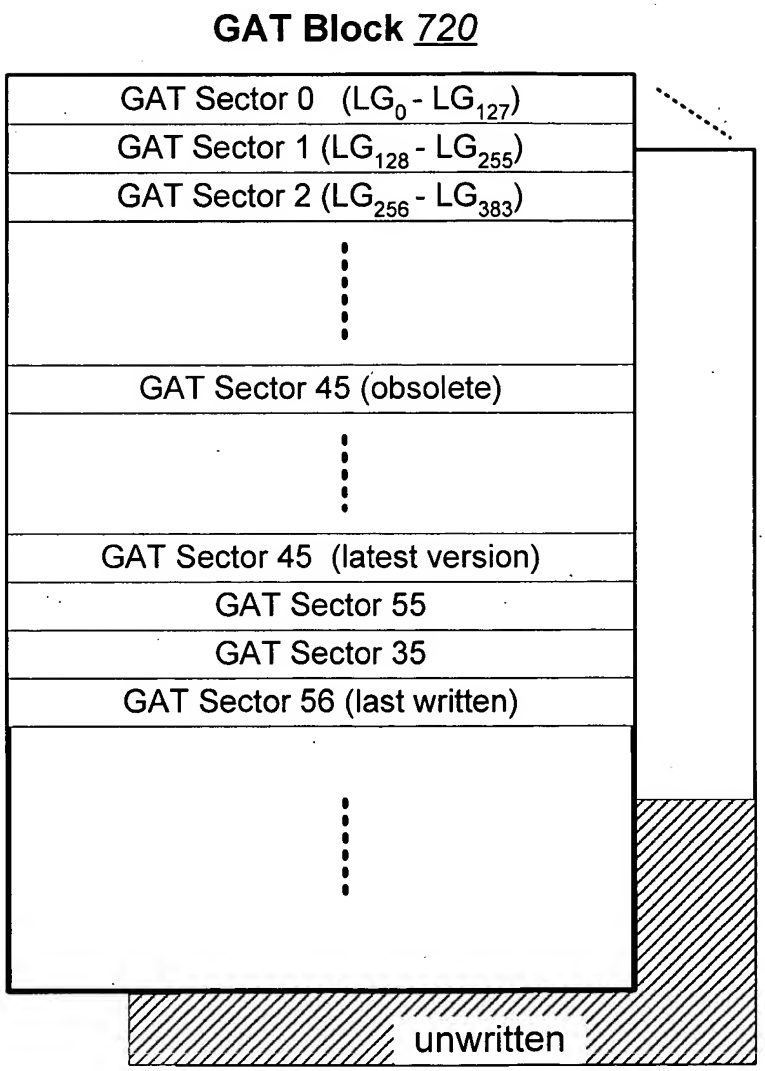


FIG. 17B

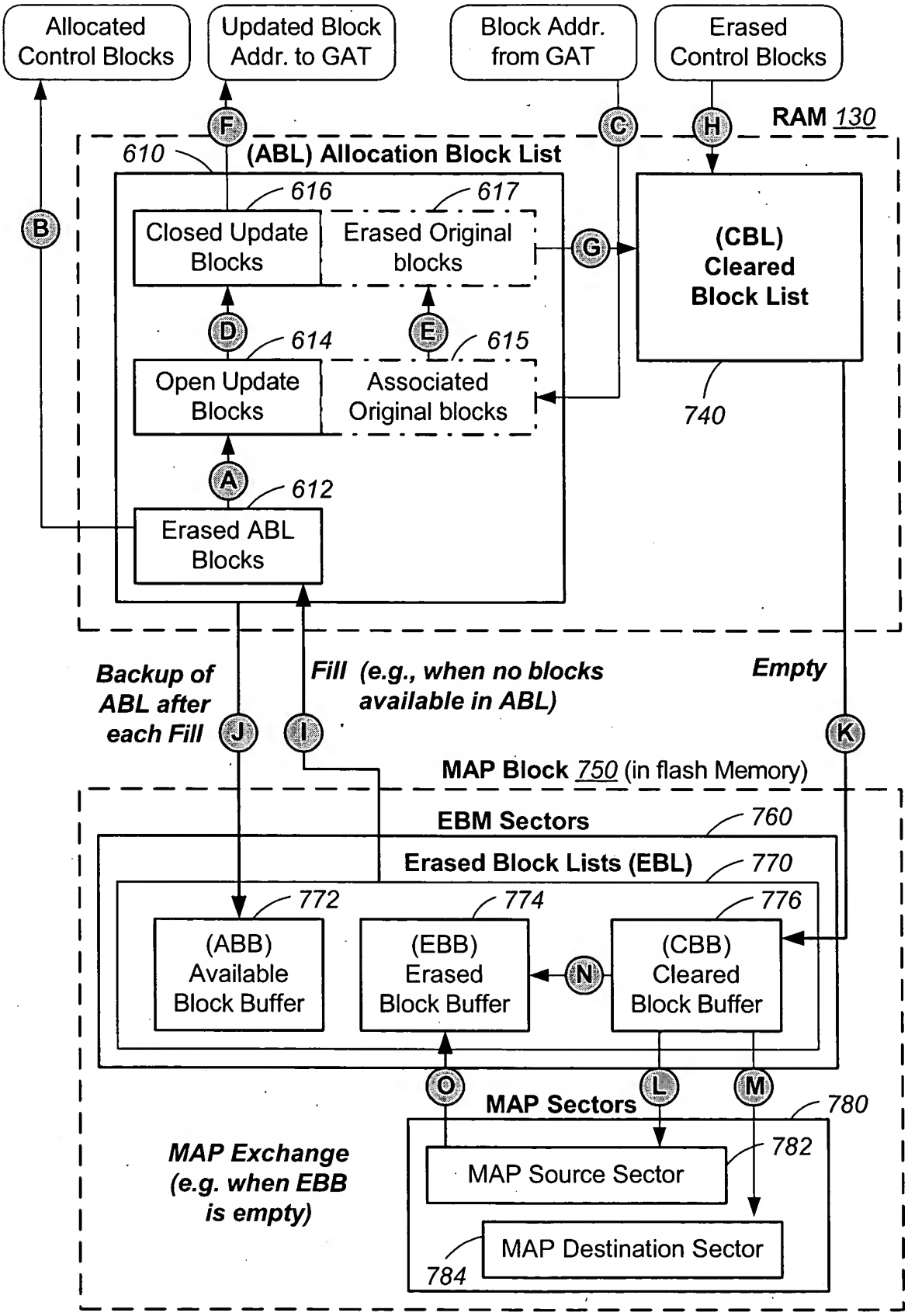
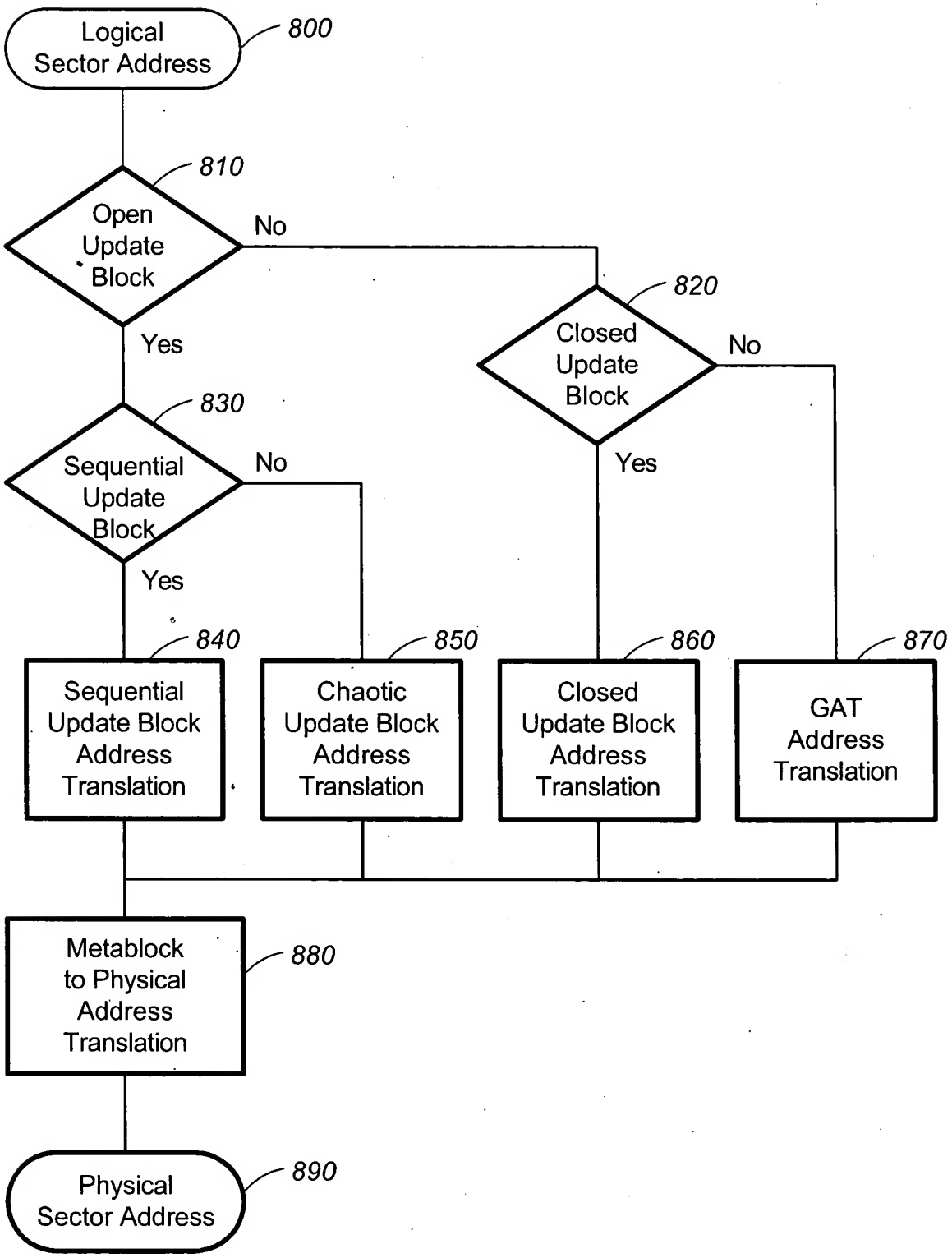
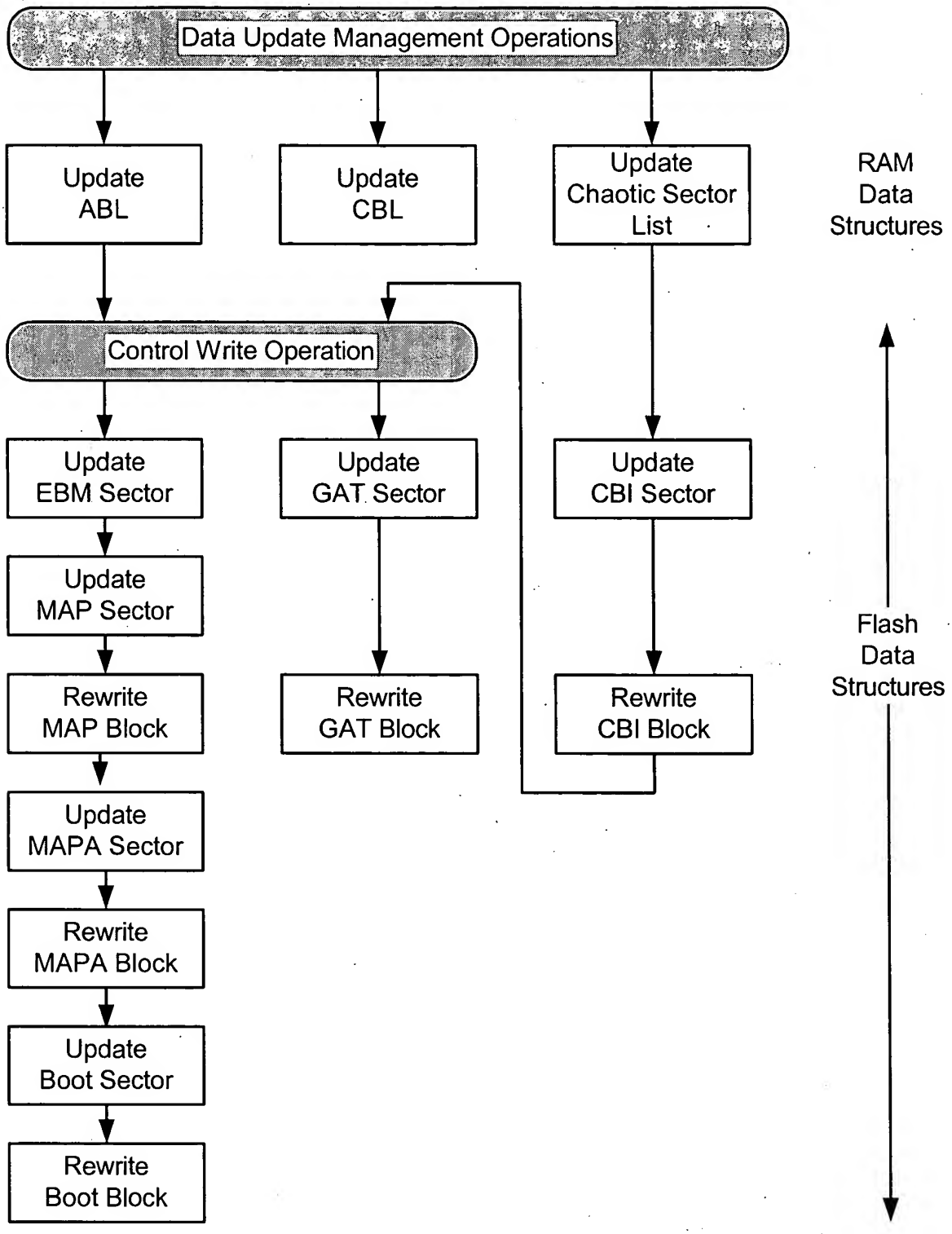

Allocated Control Blocks
Updated Block Addr. to GAT
Block Addr. from GAT
Erased Control Blocks
RAM 130
F
C
H
610
(ABL) Allocation Block List
616
617
Closed Update Blocks
Erased Original blocks
B
G
(CBL) Cleared Block List
D
E
614
615
Open Update Blocks
Associated Original blocks
740
A
612
Erased ABL Blocks
Backup of ABL after each Fill
Fill (e.g., when no blocks available in ABL)
Empty
J
I
K
MAP Block 750 (in flash Memory)
EBM Sectors
760
Erased Block Lists (EBL)
770
772
774
776
(ABB) Available Block Buffer
(EBB) Erased Block Buffer
N
(CBB) Cleared Block Buffer
O
L
M
MAP Sectors
780
MAP Exchange (e.g. when EBB is empty)
782
MAP Source Sector
MAP Destination Sector
784

FIG. 18



Address Translation

FIG. 19



Operations on Control Data Structures

FIG. 20